CLAIMS:

What is claimed is:

1. A method in a computer system, said method comprising the steps of:

executing a UNIX-based operating system within said computer system;

executing a Java desktop within said UNIX-based operating system;

executing a window manager proxy within said UNIX-based operating system;

graphically presenting native Java applications within said computer system utilizing a graphical user interface; and

graphically presenting native UNIX applications within said computer system utilizing said graphical user interface, wherein Java applications and UNIX applications are presented by said computer system utilizing the same graphical user interface.

2. The method according to claim 1, further comprising the step of distributing window manager functions between said Java desktop and said window manager proxy.

3. The method according to claim 2, wherein said step of distributing window manager functions to said Java desktop further comprises the steps of:

creating frame windows for Java applications and native UNIX applications utilizing said Java desktop;

managing user interactions with said frame windows utilizing said Java desktop; and

utilizing, by said Java desktop, said window manager

proxy to communicate with said native UNIX applications.

4. The method according to claim 3, further comprising the step of resizing, utilizing said Java desktop, frame windows for said native UNIX applications.

5. The method according to claim 3, further comprising the step of moving, utilizing said Java desktop, frame windows for said native UNIX applications.

6. The method according to claim 2, wherein the step of distributing window manager functions to said window manager proxy further comprises the steps of:

establishing a communication interface support within said window manager proxy for permitting applications to connect to and interact with said window manager proxy;

routing a first plurality of events, utilizing said window manager proxy, to said Java desktop for processing; and

processing, by said window manager proxy, a second plurality of events.

7. The method according to claim 6, wherein said step of routing a first plurality of events further comprises the steps of:

translating said first plurality of events from a first language to a second language utilizing a translator; and

forwarding said translated first plurality of events to said Java desktop.

8. The method according to claim 6, wherein said step

of translating said first plurality of events utilizing a translator further comprises the step of translating said first plurality of events utilizing a Java Native Interface.

9. The method according to claim 7, further comprising the steps of:

translating said first plurality of events from a C language to a Java language; and

forwarding said translated first plurality of events to said Java desktop.

10. The method according to claim 1, further comprising the steps of:

intercepting from one of said native UNIX applications, utilizing said window manager proxy, a frame window event to render a new window;

forwarding, utilizing said window manager proxy, said frame window event to a Java Native Interface;

translating said frame window event from a C language to a Java language utilizing said Java Native Interface;

transmitting said translated frame window event to said Java desktop; and

executing said translated frame window event utilizing said Java desktop, wherein said Java desktop renders said new window.

11. A computer system comprising:

a UNIX-based operating system being executed by said computer system;

a Java desktop being executed by said UNIX-based

operating system;

a window manager proxy being executed by said UNIX-based operating system;

said window manager proxy for graphically presenting native Java applications within said computer system utilizing a graphical user interface; and

said window manager proxy for graphically presenting native UNIX applications within said computer system utilizing said graphical user interface, wherein Java applications and UNIX applications are presented by said computer system utilizing the same graphical user interface.

12. The system according to claim 11, further comprising said Java desktop and said window manager proxy for processing window manager functions.

13. The system according to claim 12, further comprising:

said Java desktop for creating frame windows for Java applications and native UNIX applications;

said Java desktop for managing user interactions with said frame windows; and

said Java desktop for utilizing said window manager proxy to communicate with said native UNIX applications.

14. The system according to claim 13, further comprising said Java desktop for resizing frame windows for said native UNIX applications.

15. The system according to claim 13, further comprising said Java desktop for moving frame windows for said native UNIX applications.

16. The system according to claim 12, further comprising:

said window manager proxy for establishing a communication interface support within said window manager proxy for permitting applications to connect to and interact with said window manager proxy;

said window manager proxy for routing a first plurality of events to said Java desktop for processing; and

said window manager proxy for processing a second plurality of events.

17. The system according to claim 16, further comprising:

a translator for translating said first plurality of events from a first language to a second language; and

said translator for forwarding said translated first plurality of events to said Java desktop.

18. The system according to claim 16, further comprising a Java Native Interface for translating said first plurality of events.

19. The system according to claim 17, further comprising:

a translator for translating said first plurality of events from a C language to a Java language; and

said translator for forwarding said translated first plurality of events to said Java desktop.

20. The system according to claim 11, further comprising:

said window manager proxy for intercepting from one of said native UNIX applications a frame window event to render a new window;

said window manager proxy for forwarding said frame window event to a Java Native Interface;

said Java Native Interface for translating said frame window event from a C language to a Java language utilizing;

said Java Native Interface for transmitting said translated frame window event to said Java desktop; and

said Java desktop for executing said translated frame window event, wherein said Java desktop renders said new window.

21. A computer program product in a computer system, said computer program product comprising:

instruction means for executing a UNIX-based operating system within said computer system;

instruction means for executing a Java desktop within said UNIX-based operating system;

instruction means for executing a window manager proxy within said UNIX-based operating system;

instruction means for graphically presenting native Java applications within said computer system utilizing a graphical user interface; and

instruction means for graphically presenting native UNIX applications within said computer system utilizing said graphical user interface, wherein Java applications and UNIX applications are presented by said computer system utilizing the same graphical user interface.

22. The product according to claim 21, further comprising instruction means for distributing window

manager functions between said Java desktop and said window manager proxy.

23. The product according to claim 22, wherein said instruction means for distributing window manager functions to said Java desktop further comprises:

instruction means for creating frame windows for Java applications and native UNIX applications utilizing said Java desktop;

instruction means for managing user interactions with said frame windows utilizing said Java desktop; and

instruction means for utilizing, by said Java desktop, said window manager proxy to communicate with said native UNIX applications.

24. The product according to claim 23, further comprising instruction means for resizing, utilizing said Java desktop, frame windows for said native UNIX applications.

25. The product according to claim 23, further comprising instruction means for moving, utilizing said Java desktop, frame windows for said native UNIX applications.

26. The product according to claim 22, wherein said instruction means for distributing window manager functions to said window manager proxy further comprises:

instruction means for establishing a communication interface support within said window manager proxy for permitting applications to connect to and interact with said window manager proxy;

instruction means for routing a first plurality of events, utilizing said window manager proxy, to said Java desktop for processing; and

instruction means for processing, by said window manager proxy, a second plurality of events.

27. The product according to claim 26, wherein said instruction means for routing a first plurality of events further comprises:

instruction means for translating said first plurality of events from a first language to a second language utilizing a translator; and

instruction means for forwarding said translated first plurality of events to said Java desktop.

28. The product according to claim 26, wherein said instruction means for translating said first plurality of events utilizing a translator further comprises instruction means for translating said first plurality of events utilizing a Java Native Interface.

29. The product according to claim 27, further comprising:

instruction means for translating said first plurality of events from a C language to a Java language; and

instruction means for forwarding said translated first plurality of events to said Java desktop.

30. The product according to claim 21, further comprising:

instruction means for intercepting from one of said native UNIX applications, utilizing said window manager

proxy, a frame window event to render a new window;

instruction means for forwarding, utilizing said window manager proxy, said frame window event to a Java Native Interface;

instruction means for translating said frame window event from a C language to a Java language utilizing said Java Native Interface;

instruction means for transmitting said translated frame window event to said Java desktop; and

instruction means for executing said translated frame window event utilizing said Java desktop, wherein said Java desktop renders said new window.

31. A method in a computer system, said method comprising the steps of:

graphically presenting native Java applications within said computer system utilizing a graphical user interface; and

graphically presenting native UNIX applications within said computer system utilizing said graphical user interface, wherein Java applications and UNIX applications are presented by said computer system utilizing the same graphical user interface.

32. A computer system comprising:

said computer system for graphically presenting native Java applications within said computer system utilizing a graphical user interface; and

said computer system for graphically presenting

native UNIX applications within said computer system utilizing said graphical user interface, wherein Java applications and UNIX applications are presented by said computer system utilizing the same graphical user interface.

33. A computer program product in a computer system, comprising:

instruction means for graphically presenting native Java applications within said computer system utilizing a graphical user interface; and

instruction means for graphically presenting native UNIX applications within said computer system utilizing said graphical user interface, wherein Java applications and UNIX applications are presented by said computer system utilizing the same graphical user interface.